Electronic Colloquium on Computational Complexity, Report No. 190 (2026)

Efficient Randomized Communication Without Large Monochromatic Rectangles

Haoyu Wang
Penn State University
hjw5492@psu.edu

Pei Wu
Penn State University
pei.wu@psu.edu

Abstract

In this paper, we construct a total Boolean function with $\widetilde{O}(\log n)$ randomized communication protocol, while any monochromatic rectangle has density at most $O(2^{-\mathrm{poly}(n)})$. As a corollary, it gives the first total function separation for $\mathsf{BPP} \not\subseteq \mathsf{P}^{\mathsf{NP}}$ in the communication world. Inspired by Gavinsky's recent work (arXiv:2608.18784), our construction combines the cheat-sheet framework with fully linear PCPs.

1 Introduction

In the two-party communication model of Yao [Yao79], Alice and Bob receive inputs $x, y \in \{0,1\}^n$, respectively, and jointly compute a Boolean function $F(x,y)$. The goal is to communicate as few bits as possible. The randomized class BPP in communication complexity consists of families of functions computable using public randomness and at most $\mathrm{polylog}\, n$ bits of communication, with error at most $1/3$ on every input.

Understanding randomness relative to nondeterminism and the polynomial hierarchy is a central question in complexity theory. In communication complexity, BPP is incomparable with both NP and coNP [BFS86], whereas the corresponding computational comparisons remain open. At the second level of the polynomial hierarchy, the Sipser–Gács–Lautemann argument carries over to communication, as recorded by Babai, Frankl, and Simon [BFS86, Proposition 7.8]:

$$\mathsf{BPP} \subseteq \Sigma_2 \cap \Pi_2.$$

This containment is strict, since $\mathsf{NP} \subseteq \Sigma_2 \cap \Pi_2$ and $\mathsf{NP} \not\subseteq \mathsf{BPP}$. This picture motivates a finer comparison within $\Sigma_2 \cap \Pi_2$, between BPP and P^{NP}, the class of efficient deterministic protocols with adaptive access to NP predicates. These NP queries return exact answers, which can determine the choice of later queries. The question of whether $\mathsf{BPP} \subseteq \mathsf{P}^{\mathsf{NP}}$ holds for total functions was explicitly raised by Göös, Pitassi, and Watson [GPW18]. Hatami and Hatami [HH24] conjecture that this containment fails.

A second motivation is to relate randomized communication cost to exact combinatorial structure in Boolean matrices. Let $\mathsf{R}(F)$ denote the minimum cost of a public-coin randomized protocol that computes F with error at most $1/3$ on every input. View F as a matrix whose rows and columns are indexed by Alice's and Bob's inputs. A monochromatic rectangle is a product set $A \times B$, with $A, B \subseteq \{0,1\}^n$, on which F is constant. Write

$$\mathsf{rect}(F) = \max\left\{ \frac{|A||B|}{2^{2n}} : F \text{ is constant on } A \times B \right\}.$$

ISSN 1433-8092

Does every Boolean function with an efficient randomized protocol contain a large monochromatic rectangle? The log-rank conjecture provides a familiar example of such a structural question. Nisan and Wigderson [NW95] showed that this conjecture is equivalent to the universal assertion $\mathsf{rect}(F) \geq 2^{-(\mathrm{polylog}(\mathrm{rank}(F)+2))}$, where $\mathrm{rank}(F)$ is the real rank of the communication matrix. Following their separation of BPP from deterministic protocols with adaptive Equality queries, Chattopadhyay, Lovett, and Vinyals [CLV19, Problem 6.1] asked the analogous question for randomized protocols: does $\mathsf{rect}(F) \geq 2^{-O(\mathsf{R}(F))}$ always hold? Hatami and Hatami [HH24] asked this question again in their recent survey.

A structural property of P^{NP} connects these questions. Papakonstantinou, Scheder, and Song [PSS14] characterized P^{NP} through *rectangle decision lists*, which they called *rectangle overlays*. A list consists of labeled rectangles covering the input domain, and each input receives the label of the first rectangle containing it. The logarithm of the minimum list length is polynomially related to the cost of an optimal P^{NP} protocol. The *product method* of Impagliazzo and Williams [IW10] provides a lower-bound criterion for this model: every protocol of cost c has a monochromatic rectangle of measure at least $2^{-O(c)}$ under every product distribution on the two inputs. Taking the uniform distribution shows that every function in P^{NP} satisfies $\mathsf{rect}(F) \geq 2^{-\operatorname{polylog} n}$.

We construct a *total* Boolean function with small randomized communication complexity and no large monochromatic rectangle, resolving both questions.

1.1 Our results

Theorem 1.1. *There is a family of total Boolean functions $F_n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ satisfying*

$$\mathsf{R}(F_n) = O(\log n \log\log n) \qquad \textit{and} \qquad \mathsf{rect}(F_n) \leq 2^{-n^{\Omega(1)}}.$$

The family is obtained from the cheat-sheet function framework on the Gap-Hamming problem $\widetilde{\mathsf{GapHam}}$ constructed in Section 3.

The totality requirement is essential to the class separation. Gap-Hamming with a gap linear in its input length already separates randomized communication from adaptive NP queries on a promise domain: random sampling gives a constant-cost protocol, whereas the lower bounds of Papakonstantinou, Scheder, and Song [PSS14] apply to every total extension. The difficulty is to define the function outside its promise while retaining an efficient randomized protocol.

Write $\mathsf{P}^{\mathsf{NP}cc}(F)$ for the minimum cost of a deterministic protocol with adaptive NP queries, charged their nondeterministic communication cost. Precise cost conventions appear in Section 2. The product method of Impagliazzo and Williams [IW10] gives

$$\mathsf{P}^{\mathsf{NP}cc}(F) = \Omega\left(\log \frac{1}{\mathsf{rect}(F)}\right).$$

Applying this bound under the uniform distribution to Theorem 1.1 yields the following separation.

Corollary 1.2. *The family $\{F_n\}$ in Theorem 1.1 satisfies*

$$\mathsf{R}(F_n) = O(\log n \log\log n) \qquad \textit{and} \qquad \mathsf{P}^{\mathsf{NP}cc}(F_n) \geq n^{\Omega(1)}.$$

In particular, $\mathsf{BPP} \not\subseteq \mathsf{P}^{\mathsf{NP}}$ for total functions.

Since $\mathsf{NP} \not\subseteq \mathsf{BPP}$ was already known [BFS86], Corollary 1.2 establishes that BPP and P^{NP} are incomparable within $\Sigma_2 \cap \Pi_2$, even for total functions.

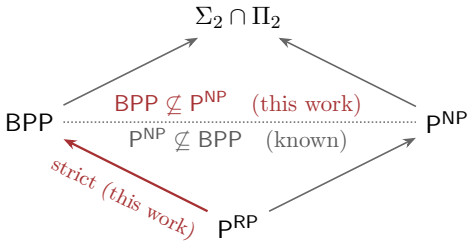

Figure 1: Selected communication classes on total functions; new separations are highlighted in red.

The second corollary concerns RP, the one-sided-error version of BPP: protocols never err on 0-inputs and have error at most $1/3$ on 1-inputs. We have the standard containment

$$\mathsf{P}^{\mathsf{RP}} \subseteq \mathsf{BPP} \cap \mathsf{P}^{\mathsf{NP}}.$$

Indeed, error amplification allows efficient adaptive RP-query protocols to be simulated in BPP, while $\mathsf{RP} \subseteq \mathsf{NP}$ gives the containment in P^{NP} [HH24]. Write $\mathsf{P}^{\mathsf{RP}cc}(F)$ for the analogous cost with exact RP queries, charged their one-sided randomized communication cost. Since an RP query of cost $h \geq 1$ has nondeterministic cost at most $h + O(\log n)$ [HH24], simulating each query gives $\mathsf{P}^{\mathsf{NP}cc}(F) \leq O(\log n)\,\mathsf{P}^{\mathsf{RP}cc}(F)$. Corollary 1.2 then gives the following quantitative consequence.

Corollary 1.3. *The family $\{F_n\}$ in Theorem 1.1 satisfies*

$$\mathsf{R}(F_n) = O(\log n \log\log n) \qquad \textit{and} \qquad \mathsf{P}^{\mathsf{RP}cc}(F_n) \geq n^{\Omega(1)}.$$

In particular, $\mathsf{P}^{\mathsf{RP}} \subsetneq \mathsf{BPP}$ for total functions.

Corollary 1.3 shows that even adaptive composition of efficient one-sided-error randomized protocols cannot simulate every efficient two-sided-error randomized protocol, answering a question of Pitassi, Shirley, and Watson [PSW20]. Figure 1 summarizes these relations.

1.2 Proof overview

Our construction uses the cheat-sheet method of Aaronson, Ben-David, and Kothari [ABDK16], introduced in query complexity to obtain separations for total functions from promise problems. Anshu et al. [ABBD$^+$16] developed the method in communication complexity.

We use Gap-Hamming with a linear gap as the outer gadget: its outputs address a table of certificates, the cheat-sheet. The inner gadget uses fully linear PCPs[1] to certify that the base input satisfies the Gap-Hamming domain promises. Gavinsky [Gav26] recently used this inner gadget to separate quantum and randomized communication for a total function.

At a high level, the randomized upper bound is from random sampling for Gap-Hamming and the verification guarantee of fully linear PCPs. The remaining task is to show that the resulting total function has no large monochromatic rectangle.

Fix an integer k, let $m = 2^k$, and put $d = O(km^2)$. Alice and Bob receive inputs

$$(\mathbf{x}, \mathbf{u}),\ (\mathbf{y}, \mathbf{v}) \in \{0,1\}^{k \cdot m} \times \{0,1\}^{m \cdot 4k(2d-1)},$$

[1]We remark here, the fully linear PCP of Boneh et al. is designed for secret-shared inputs. Its verifier accesses the input and proof jointly through a constant number of linear functionals; Boneh et al. call such a system a fully linear PCP. It has no implication to the proof length.

respectively. The strings $\mathbf{x}$ and $\mathbf{y}$ are the inputs to k Gap-Hamming instances, each on m bits per party. The strings $\mathbf{u}$ and $\mathbf{v}$ are the cheat-sheet inputs: each is a table of m entries, with $4k(2d-1)$ bits per entry. Thus each party's input has length $n = km + 4km(2d-1)$, so $k = \Theta(\log n)$ and $m = n^{\Theta(1)}$. This construction is illustrated in Figure 2.

Gap-Hamming supplies an address. Write $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_k)$ and $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_k)$, where $\mathbf{x}_i, \mathbf{y}_i \in \{0,1\}^m$, and put $\mathbf{z} = \mathbf{x} \oplus \mathbf{y}$. For each pair, $\mathsf{GapHam}(\mathbf{x}_i, \mathbf{y}_i)$ is 0 when $|\mathbf{z}_i| \le m/3$, is 1 when $|\mathbf{z}_i| \ge 2m/3$, and is undefined otherwise. The thresholds are separated by $m/3$. Equivalently, each instance applies gap majority to the XOR block $\mathbf{z}_i$. The indicator $\mathsf{Ind}(\mathbf{z})$ records whether all k pairs satisfy these promises. When $\mathsf{Ind}(\mathbf{z}) = 1$, their k answers specify an address $\mathsf{Ads}(\mathbf{z}) \in [m]$, using the encoding in Section 3.

The linear gap makes this address inexpensive to estimate. To make all k address bits correct with constant probability, we reduce the error on each instance to $O(1/k)$. This requires $O(\log k)$ sampled coordinates per instance and hence $O(k \log k)$ communication.

Fully linear PCPs supply the certificates. We use the fully linear PCP construction of Boneh et al. [BBCG$^+$19] to certify that all Gap-Hamming instances satisfy their promises. For each $\mathbf{z}$, a string $\mathbf{c} \in \{0,1\}^{4k(2d-1)}$ is a proposed certificate for $\mathsf{Ind}(\mathbf{z}) = 1$. We define the certificate predicate $\mathsf{Check}(\mathbf{z}, \mathbf{c})$ so that an accepting certificate exists if and only if $\mathsf{Ind}(\mathbf{z}) = 1$.

A fully linear PCP is a probabilistically checkable proof whose verifier accesses both the input and the proof through linear functionals over a finite field. The construction we use requires only $O(1)$ such queries.

This query model is particularly useful when the input and proof are shared between Alice and Bob. Suppose $\mathbf{c} = \mathbf{c}_A \oplus \mathbf{c}_B$. Over $\mathbb{F}_{2^{4k}}$, the XOR shares become additive shares, so for any linear functional L,

$$L(\mathbf{z}, \mathbf{c}) = L(\mathbf{x}, \mathbf{c}_A) + L(\mathbf{y}, \mathbf{c}_B).$$

The public randomness specifies all the queries. For each one, Alice evaluates the functional on her shares, and Bob can recover the answer by adding his local value. Since the queries are nonadaptive, Alice sends all $O(1)$ field elements in one message.

The combined cheat sheet. The cheat-sheet table has one cell for each of the $m = 2^k$ possible vectors of Gap-Hamming outputs. Alice holds $\mathbf{u} = (\mathbf{u}_1, \ldots, \mathbf{u}_m)$ and Bob holds $\mathbf{v} = (\mathbf{v}_1, \ldots, \mathbf{v}_m)$. At entry j, their two shares specify the proposed certificate

$$\mathbf{c}_j = \mathbf{u}_j \oplus \mathbf{v}_j.$$

When the base input satisfies every promise, its k Gap-Hamming outputs select one cell, and the function checks only the certificate in that cell. Writing $h = \mathsf{Ads}(\mathbf{z})$, we define

$$\widetilde{\mathsf{GapHam}}(\mathbf{x}, \mathbf{u}; \mathbf{y}, \mathbf{v}) = \begin{cases} \mathsf{Check}(\mathbf{z}, \mathbf{c}_h), & \text{if } \mathsf{Ind}(\mathbf{z}) = 1, \\ 0, & \text{if } \mathsf{Ind}(\mathbf{z}) = 0. \end{cases}$$

This definition covers every input. If some Gap-Hamming instance violates its promise, the output is 0 independently of the table. If all instances satisfy their promises, every cell except the ℓ-th is ignored, and the output is 1 exactly when $\mathbf{c}_\ell$ is the unique valid certificate. The table is unrestricted, so an invalid certificate at the selected cell produces output 0. Figure 2 depicts the combined construction.

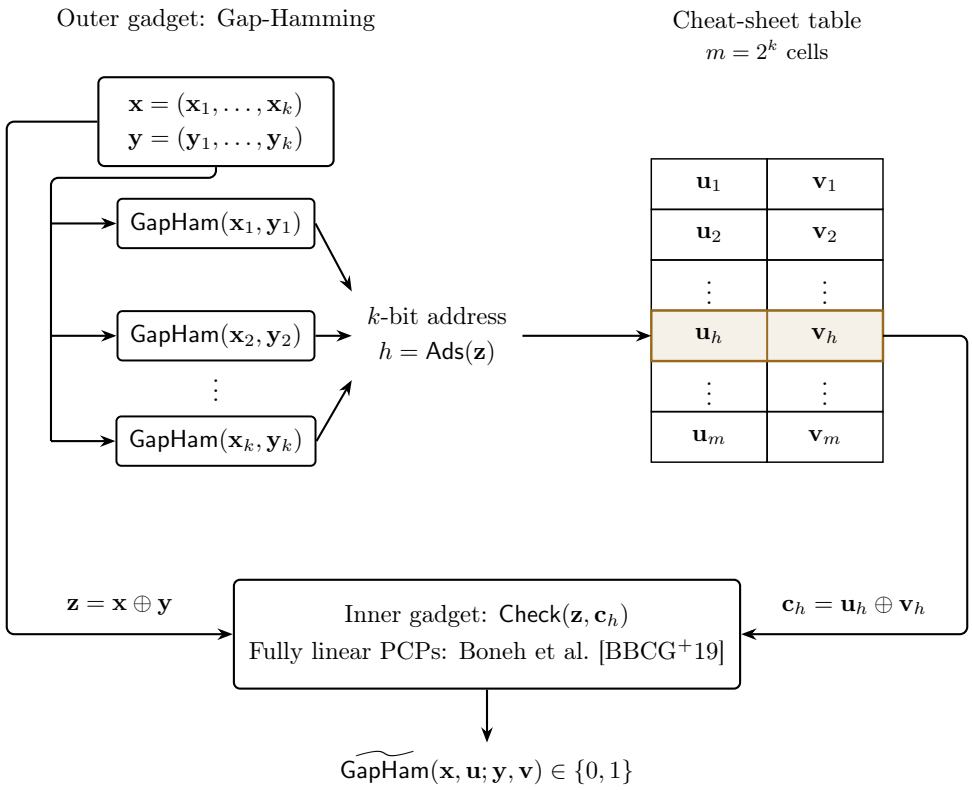

Figure 2: The Gap-Hamming cheat-sheet construction on promised base inputs. The k Gap-Hamming outputs select one of $m = 2^k$ entries in the XOR-shared certificate table. The selected certificate is checked against the base input using fully linear PCPs of Boneh et al. [BBCG$^+$19]; the inner predicate tests whether this certificate proves $\mathsf{Ind}(\mathbf{z}) = 1$.

Excluding large monochromatic rectangles. Ruling out a large 0-rectangle is delicate and involves several steps. Given a somewhat large rectangle, we find a 1-input inside it as follows. First, we retain base inputs on each side that have many compatible tables; the resulting sets of base inputs remain large. Second, a counting argument shows that each retained base input has only a few addresses at which at most half of all table-entry values occur. Third, double counting gives an address h for which more than half of all values occur for many retained inputs on both sides. We restrict to these inputs, obtaining two large sets X_h and Y_h. Fourth, the Hamming-distance argument finds $\mathbf{x} \in X_h$ and $\mathbf{y} \in Y_h$ whose Gap-Hamming outputs select address h. Finally, let $\mathbf{c}^*$ be the valid certificate for this pair. Because $\mathbf{x} \in X_h$, more than half of all certificate shares occur as the h-th entry of a table compatible with $\mathbf{x}$ on Alice's side; the analogous statement holds for $\mathbf{y} \in Y_h$ on Bob's side. After translating Bob's set of shares by $\mathbf{c}^*$, these two sets still contain more than half of all shares and therefore intersect. This gives compatible tables whose h-th entries have XOR $\mathbf{c}^*$, producing a 1-input in the proposed 0-rectangle. Large 1-rectangles are ruled out directly by the density of 1-inputs.

2 Preliminaries

Notation. For a positive integer n, write $[n] = \{1, \ldots, n\}$. All logarithms are to base 2, except that ln denotes the natural logarithm. We write $\operatorname{polylog} n$ for $(\log n)^{O(1)}$. For a binary string $\mathbf{z}$,

write $\mathbf{z}(j)$ for its j-th coordinate and $|\mathbf{z}|$ for its Hamming weight. When a string is partitioned into blocks, subscripts index the blocks, so $\mathbf{z}_i(j)$ is the j-th coordinate of block $\mathbf{z}_i$. The operation $\oplus$ denotes bitwise XOR, so $|\mathbf{x} \oplus \mathbf{y}|$ is the Hamming distance. For an event E, its indicator $\mathbf{1}\{E\}$ is 1 when E holds and 0 otherwise. The density of a subset S of a finite nonempty set Ω is $|S|/|\Omega|$.

Rectangles. Let $F : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ be a total Boolean function on finite nonempty input sets. A rectangle is a set $A \times B$ with $A \subseteq \mathcal{X}$ and $B \subseteq \mathcal{Y}$. It is a monochromatic b-rectangle if $F(x,y) = b$ for every $(x,y) \in A \times B$, $b \in \{0,1\}$. We write

$$\mathsf{rect}(F) = \max\left\{ \frac{|A||B|}{|\mathcal{X}||\mathcal{Y}|} : A \times B \text{ is monochromatic for } F \right\}.$$

Thus $\mathsf{rect}(F)$ is the largest monochromatic rectangle density under the uniform distribution. For a product probability measure $\mu = \mu_{\mathcal{X}} \times \mu_{\mathcal{Y}}$, we have $\mu(A \times B) := \mu_{\mathcal{X}}(A)\mu_{\mathcal{Y}}(B)$.

Communication models. Alice receives $x \in \mathcal{X}$ and Bob receives $y \in \mathcal{Y}$. Write $\mathsf{R}(F)$ for the minimum communication cost of a public-coin randomized protocol computing F with error at most $1/3$ on every input. Its cost is the worst-case communication cost over all inputs and random bits.

For $\mathsf{P}^{\mathsf{NP}cc}(F)$, we use the model of Göös et al. [GKPW17]. A protocol is a deterministic binary decision tree whose leaves are labeled by output bits. At each internal node, the parties receive the exact value of a predicate $Q(x,y)$. A query has cost $h \geq 1$ if its 1-set can be covered by at most 2^h rectangles. The answer determines the next node, so queries may depend on earlier answers. The cost of the protocol is the maximum, over all inputs, of the sum of query costs along the resulting path. We write $\mathsf{P}^{\mathsf{NP}cc}(F)$ for the minimum cost of a protocol computing F exactly. Ordinary communication is included: sending one bit is a query of cost 1. Figure 3 illustrates the rectangle covers and adaptive queries in this model.

Following Hatami and Hatami [HH24], define $\mathsf{P}^{\mathsf{RP}cc}(F)$ in the same way, with each query charged its public-coin one-sided randomized communication complexity. Such a randomized protocol must output 0 whenever $Q(x,y) = 0$ and must output 1 with probability at least $2/3$ whenever $Q(x,y) = 1$. We impose a minimum query cost of 1, as in the NP-query model. The oracle still returns $Q(x,y)$ exactly; randomization enters only in the cost assigned to the query.

For families $\{F_n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}\}$, the classes BPP, P^{NP}, and P^{RP} consist of those with respective costs $\mathsf{R}(F_n)$, $\mathsf{P}^{\mathsf{NP}cc}(F_n)$, and $\mathsf{P}^{\mathsf{RP}cc}(F_n)$ bounded by $(\log n)^{O(1)}$. All complexity classes in this paper refer to communication complexity.

The product method. We use the following lower bound of Impagliazzo and Williams [IW10]; see also [GKPW17].

Fact 2.1 (Impagliazzo-Williams)**.** *Let μ be a product probability measure on $\mathcal{X} \times \mathcal{Y}$. If every monochromatic rectangle of F has μ-measure at most δ, where $0 < \delta \leq 1$, then $\mathsf{P}^{\mathsf{NP}cc}(F) = \Omega(\log(1/\delta))$. In particular,*

$$\mathsf{P}^{\mathsf{NP}cc}(F) = \Omega\left(\log \frac{1}{\mathsf{rect}(F)}\right).$$

3 Totalization via cheat sheets

We extend the Gap-Hamming communication problem to a total function by adding tables of certificates, following the cheat-sheet construction [ABBD$^+$16, ABDK16]. Throughout the construction, fix an integer $k \geq 12$ and let $m = 2^k$.

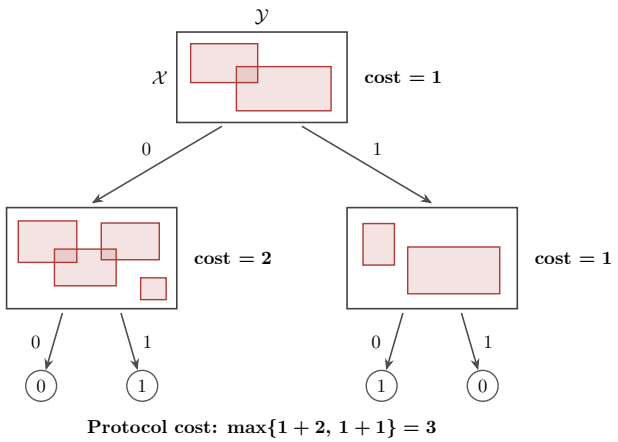

Figure 3: An adaptive P^{NP} protocol in the model of Göös et al. [GKPW17]. Each box depicts a query on $\mathcal{X} \times \mathcal{Y}$: the shaded rectangles cover its 1-set, and the white region is its 0-set. A cover by at most 2^h rectangles permits a query of cost $h \geq 1$. The exact answer selects the next query; leaves give the final output.

Definition 3.1. For $\mathbf{a}, \mathbf{b} \in \{0,1\}^m$, the Gap-Hamming function is

$$\mathsf{GapHam}(\mathbf{a}, \mathbf{b}) = \begin{cases} 0, & |\mathbf{a} \oplus \mathbf{b}| \leq m/3, \\ 1, & |\mathbf{a} \oplus \mathbf{b}| \geq 2m/3, \\ *, & m/3 < |\mathbf{a} \oplus \mathbf{b}| < 2m/3. \end{cases}$$

The symbol $*$ means that the output is undefined.

Define

$$\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_k), \qquad \mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_k), \qquad \mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_k) = \mathbf{x} \oplus \mathbf{y},$$

where $\mathbf{x}_i, \mathbf{y}_i \in \{0,1\}^m$ and $\mathbf{z}_i = \mathbf{x}_i \oplus \mathbf{y}_i$. For each $h \in [m]$, let $(h_1, h_2, \ldots, h_k) \in \{0,1\}^k$ be the binary expansion, i.e.,

$$h = 1 + \sum_{i=1}^{k} h_i 2^{k-i}.$$

Thus the outputs of k Gap-Hamming instances specify an address in the table.

Definition 3.2. For $\mathbf{z} \in \{0,1\}^{km}$, define the indicator function as

$$\mathsf{Ind}(\mathbf{z}) = \prod_{i=1}^{k} \mathbf{1}\big\{|\mathbf{z}_i| \leq m/3 \text{ or } |\mathbf{z}_i| \geq 2m/3\big\}.$$

If GapHam is well defined on all $(\mathbf{x}_i, \mathbf{y}_i)$, i.e., $\mathsf{Ind}(\mathbf{z}) = 1$, we obtain an address

$$\mathsf{Ads}(\mathbf{z}) := 1 + \sum_{i=1}^{k} \mathsf{GapHam}(\mathbf{x}_i, \mathbf{y}_i) \cdot 2^{k-i} \in [m].$$

Otherwise, define $\mathsf{Ads}(\mathbf{z})$ arbitrarily in $[m]$.

3.1 A circuit for $1 - \mathsf{Ind}(\mathbf{z})$.

In the cheat sheet totalization construction, the fully linear PCP from Boneh et al. [BBCG+19] serves to checks whether all Gap-Hamming inputs satisfy their promises. The PCP builds on a circuit with certificate; the circuit computes $1 - \mathsf{Ind}(\mathbf{z})$; and the certificate claims the circuit will output 0, which is equivalent to $\mathsf{Ind}(\mathbf{z}) = 1$. Alice and Bob can efficiently check this claim by exchanging $O(k)$ bits. We first describe the circuit.

Lemma 3.3. *There is a circuit using fan-in two* XOR *and* AND *gates,* NOT *gates, and constants* $\{0,1\}$*, that computes* $1-\mathsf{Ind}(\mathbf{z})$ *for every* $\mathbf{z} \in \{0,1\}^{km}$*. Furthermore, it includes* $d = k\big(m(m+3)+1\big)$ AND *gates.*

For each $i \in [k]$ and $0 \le j \le m$, define $b^{(i)}_{j,r} = 0$ whenever $r < 0$ or $r > j$, and set $b^{(i)}_{0,0} = 1$. For $1 \le j \le m$ and $0 \le r \le j$, given the input $\mathbf{z}$, compute

$$b^{(i)}_{j,r} = \big(\neg \mathbf{z}_i(j) \wedge b^{(i)}_{j-1,r}\big) \oplus \big(\mathbf{z}_i(j) \wedge b^{(i)}_{j-1,r-1}\big). \tag{3.1}$$

The following identity shows that these bits record the number of ones in each prefix segment of $\mathbf{z}_i$.

Claim 3.4. *For each* $i \in [k]$*,* $0 \le j \le m$*, and* $0 \le r \le j$*,*

$$b^{(i)}_{j,r} = \mathbf{1}\Big\{ \sum_{t=1}^{j} \mathbf{z}_i(t) = r \Big\}.$$

Proof. We prove the identity by induction on j. For $j = 0$, the sum is empty and $b^{(i)}_{0,0} = 1$. Suppose the identity holds for $j-1$. By the zero values fixed above, it also holds for $r < 0$ or $r > j-1$.

If $\mathbf{z}_i(j) = 0$, then (3.1) gives

$$b^{(i)}_{j,r} = b^{(i)}_{j-1,r} = \mathbf{1}\Big\{ \sum_{t=1}^{j-1} \mathbf{z}_i(t) = r \Big\} = \mathbf{1}\Big\{ \sum_{t=1}^{j} \mathbf{z}_i(t) = r \Big\}.$$

If $\mathbf{z}_i(j) = 1$, then (3.1) gives

$$b^{(i)}_{j,r} = b^{(i)}_{j-1,r-1} = \mathbf{1}\Big\{ \sum_{t=1}^{j-1} \mathbf{z}_i(t) = r-1 \Big\} = \mathbf{1}\Big\{ \sum_{t=1}^{j} \mathbf{z}_i(t) = r \Big\}.$$

These two cases prove the identity for j. □

We describe the circuit as follows.

Proof of Lemma 3.3. Given the $\{b^{(i)}_{j,r}\}$, for each $i \in [k]$, the circuit computes

$$B_i = \bigoplus_{\substack{0 \le r \le m \\ r \le m/3 \text{ or } r \ge 2m/3}} b^{(i)}_{m,r}.$$

By Claim 3.4, exactly one of $\{b^{(i)}_{m,0}, \dots, b^{(i)}_{m,m}\}$ equals 1. Therefore $B_i = \mathbf{1}\big\{|\mathbf{z}_i| \le m/3 \text{ or } |\mathbf{z}_i| \ge 2m/3\big\}$. Then the circuit outputs

$$\neg(B_1 \wedge \cdots \wedge B_k) \wedge 1 = 1 - \prod_{i=1}^{k} B_i = 1 - \mathsf{Ind}(\mathbf{z}). \tag{3.2}$$

Specifically, compute $\neg(B_1 \wedge \cdots \wedge B_k) \wedge 1$ using AND gates with two inputs.

Each pair (j, r) in (3.1) uses two AND gates, and each j has $j+1$ choices of r. Equation (3.2) uses $k-1$ gates for $B_1 \wedge \cdots \wedge B_k$ and one final gate. So the number of AND gates is

$$2k \sum_{j=1}^{2^k} (j+1) + k = k\big(2^k(2^k+3)+1\big) = O(km^2). \qquad \square$$

In the next section, we introduce the certificate.

3.2 Circuits to certificates.

We arithmetize the circuit for use in the fully linear PCP. In particular, we will express operations of the circuit over a finite field and encode its AND outputs by a polynomial.

At first, label the AND gates according to the calculation order (bottom-up) by $1, \ldots, d$. Recall that $d = k\big(m(m+3)+1\big)$ by Lemma 3.3. We perform arithmetic in the field $\mathbb{F}_{2^{4k}}$ which has characteristic 2. More specifically, fix $e_1 = 1, e_2, \ldots, e_{4k} \in \mathbb{F}_{2^{4k}}$ such that every $a \in \mathbb{F}_{2^{4k}}$ can be written uniquely as

$$a = \sum_{i=1}^{4k} a_i e_i, \qquad a_i \in \{0,1\}.$$

These elements form a basis over $\mathbb{F}_2 = \{0,1\}$, and the coefficient string $(a_1, \ldots, a_{4k})$ encodes a. In particular, the field elements 0 and 1 are encoded by $(0, \ldots, 0)$ and $(1, 0, \ldots, 0)$. Addition in $\mathbb{F}_{2^{4k}}$ corresponds to bitwise XOR of these $4k$-bit encodings: each coordinate is added modulo 2, while the result remains a $4k$-bit encoding of a field element. We represent Boolean inputs $0, 1$ by these field elements. For Boolean a, b, the gate operations therefore satisfy

$$a \wedge b = ab, \qquad a \oplus b = a + b, \qquad \neg a = 1 + a,$$

where the operations on the right are field operations. Since $d < 2k\,4^k < 2^{4k}$, we can fix distinct

$$\tau_1, \tau_2, \ldots, \tau_d \in \mathbb{F}_{2^{4k}}.$$

Definition 3.5 (Certificate)**.** A certificate $\mathbf{c} \in \{0,1\}^{4k \cdot (2d-1)}$ consists of $2d-1$ blocks of $4k$ bits. Let $c_0, \ldots, c_{2d-2} \in \mathbb{F}_{2^{4k}}$ be the field elements encoded by these blocks, and define the polynomial in the variable T by

$$p(T) = \sum_{v=0}^{2d-2} c_v T^v.$$

Certificates and polynomials of degree at most $2d-2$ correspond one to one. The certificate claims that the v-th AND gate outputs $p(\tau_v)$; and we show how to check this claim efficiently in the next section.

3.3 The fully linear PCP

Given $\mathbf{z}$ and the polynomial p encoded by certificate $\mathbf{c}$, process the circuit in calculation order. At an XOR or NOT gate, apply the field operations above. At the v-th AND gate, use $p(\tau_v)$ directly as its output.

Let $a_v(\mathbf{z}, p)$ and $b_v(\mathbf{z}, p)$ be the left and right inputs obtained at the v-th AND gate. They depend only on $\mathbf{z}$ and the earlier claimed outputs $p(\tau_1), \ldots, p(\tau_{v-1})$. Each is a fixed weighted sum

of the coordinates of $\mathbf{z}$ and the coefficients of p, plus a constant: the only operations used between claimed outputs are addition and addition of 1. So $a_v(\mathbf{z}, p)$ and $b_v(\mathbf{z}, p)$ are affine functions on $\mathbf{z}$ and $p(\tau_1), \ldots, p(\tau_d)$.

For $v \in [d]$, define

$$\delta_v(T) = \prod_{\substack{j \in [d] \\ j \neq v}} \frac{T - \tau_j}{\tau_v - \tau_j}.$$

These polynomials satisfy $\delta_v(\tau_v) = 1$ and $\delta_v(\tau_j) = 0$ for $j \neq v$. Define

$$f_{(\mathbf{z},p)}(T) = \sum_{v=1}^{d} \delta_v(T) a_v(\mathbf{z}, p), \quad g_{(\mathbf{z},p)}(T) = \sum_{v=1}^{d} \delta_v(T) b_v(\mathbf{z}, p).$$

Their degrees are at most $d - 1$, and

$$f_{(\mathbf{z},p)}(\tau_v) = a_v(\mathbf{z}, p), \qquad g_{(\mathbf{z},p)}(\tau_v) = b_v(\mathbf{z}, p).$$

For each fixed T, both evaluations are affine functions of $\mathbf{z}$ and the coefficients of p.

We require a polynomial identity to enforce the multiplication equations, and require the final output to be 0. Define

$$\mathsf{Check}(\mathbf{z}, p) = \begin{cases} 1, & p \equiv f_{(\mathbf{z},p)} g_{(\mathbf{z},p)} \text{ and } p(\tau_d) = 0, \\ 0, & \text{otherwise.} \end{cases}$$

Let $\mathsf{Check}(\mathbf{z}, \mathbf{c}) := \mathsf{Check}(\mathbf{z}, p)$ for the polynomial p encoded by $\mathbf{c}$.

The following lemma shows that a true certificate exists exactly when $\mathsf{Ind}(\mathbf{z}) = 1$, and that the polynomial identity makes the certificate unique. For brevity, we defer the proof to Appendix A.

Lemma 3.6. *The function* $\mathsf{Check}(\mathbf{z}, p)$ *determines whether* p *certifies that the circuit output is* 0*. Moreover,*

1). If $\mathsf{Ind}(\mathbf{z}) = 1$*, exactly one certificate* $\mathbf{c} \in \{0,1\}^{4k(2d-1)}$ *satisfies* $\mathsf{Check}(\mathbf{z}, \mathbf{c}) = 1$*.*

2). If $\mathsf{Ind}(\mathbf{z}) = 0$*, every certificate* $\mathbf{c} \in \{0,1\}^{4k(2d-1)}$ *satisfies* $\mathsf{Check}(\mathbf{z}, \mathbf{c}) = 0$*.*

We show the randomized test for the polynomial condition in Check as follows. A linear query returns a weighted sum of the coordinates of $\mathbf{z}$ and the coefficients of p. Let $\mathbf{0} \in \{0,1\}^{km}$ be the zero vector, and let $p_0(T) \equiv 0$ be the zero polynomial.

Theorem 3.7 (Fully linear PCP, cf. [BBCG$^+$19])**.** *Fix* $\mathbf{z} \in \{0,1\}^{km}$ *and a polynomial* $p \in \mathbb{F}_{2^{4k}}[T]$ *of degree at most* $2d - 2$*. There is a randomized test using four linear queries: if* $\mathsf{Check}(\mathbf{z}, p) = 1$*, it always outputs* 1*; if* $\mathsf{Check}(\mathbf{z}, p) = 0$*, it outputs* 1 *with probability at most* $1/6$*.*

Proof. Choose $r \in \mathbb{F}_{2^{4k}}$ uniformly. The four queries return

$$p(r), \quad f_{(\mathbf{z},p)}(r) - f_{(\mathbf{0},p_0)}(r), \quad g_{(\mathbf{z},p)}(r) - g_{(\mathbf{0},p_0)}(r), \quad p(\tau_d).$$

These are linear functions of $\mathbf{z}$ and the coefficients of p: the second and third expressions subtract the constants from the corresponding affine functions, which depend only on the circuit and r. Adding constants $f_{(\mathbf{0},p_0)}(r)$ and $g_{(\mathbf{0},p_0)}(r)$ to the second and third answers gives $f_{(\mathbf{z},p)}(r)$ and $g_{(\mathbf{z},p)}(r)$, respectively.

The test outputs 1 if and only if

$$p(\tau_d) = 0 \quad \text{and} \quad p(r) = f_{(\mathbf{z},p)}(r)g_{(\mathbf{z},p)}(r).$$

If $\mathsf{Check}(\mathbf{z}, p) = 1$, both equalities hold for every r. Suppose $\mathsf{Check}(\mathbf{z}, p) = 0$. If $p(\tau_d) \neq 0$, the test always outputs 0. Otherwise, $p - f_{(\mathbf{z},p)}g_{(\mathbf{z},p)}$ is a nonzero polynomial of degree at most $2d - 2$. It has at most $2d - 2$ roots, so

$$\Pr_r[\text{the test outputs 1}] \leq \frac{2d-2}{2^{4k}} < \frac{1}{6}. \qquad \square$$

3.4 The total function

Now we combine the fully linear PCP and Gap-Hamming function to construct our total function. Besides $\mathbf{x} \in \{0,1\}^{km}$ and $\mathbf{y} \in \{0,1\}^{km}$, Alice and Bob each receive a table with m positions:

$$\mathbf{u} = (\mathbf{u}_1, \ldots, \mathbf{u}_m), \qquad \mathbf{v} = (\mathbf{v}_1, \ldots, \mathbf{v}_m),$$

where $\mathbf{u}_j, \mathbf{v}_j \in \{0,1\}^{4k(2d-1)}$. At position j, the two table entries together specify the certificate

$$\mathbf{c}_j = \mathbf{u}_j \oplus \mathbf{v}_j.$$

The total function uses the certificate at the address determined by the Gap-Hamming outputs.

Definition 3.8 (Gap-Hamming with cheat sheets)**.** For inputs $(\mathbf{x}, \mathbf{u})$ and $(\mathbf{y}, \mathbf{v})$ in $\{0,1\}^{km} \times \{0,1\}^{4km(2d-1)}$, let $\mathbf{z} = \mathbf{x} \oplus \mathbf{y}$ and $h = \mathsf{Ads}(\mathbf{z}) \in [m]$. Define

$$\widetilde{\mathsf{GapHam}}(\mathbf{x}, \mathbf{u}; \mathbf{y}, \mathbf{v}) = \begin{cases} \mathsf{Check}(\mathbf{z}, \mathbf{c}_h), & \mathsf{Ind}(\mathbf{z}) = 1, \\ 0, & \mathsf{Ind}(\mathbf{z}) = 0. \end{cases}$$

Each player receives an input of length $n = km + 4km(2d-1)$. Substituting the value of d gives $k = \Theta(\log n)$. For the two-party communication model, Alice has $(\mathbf{x}, \mathbf{u})$ and Bob has $(\mathbf{y}, \mathbf{v})$; we give a randomized protocol on $\widetilde{\mathsf{GapHam}}$ with cost $O(\log n \log\log n)$ in the next section.

4 An Efficient Randomized Protocol

To compute $\widetilde{\mathsf{GapHam}}$, Alice and Bob first estimate an address by a random sketch of inputs $(\mathbf{x}, \mathbf{y})$. Then they apply the test from Theorem 3.7 to estimate the Check function.

Theorem 4.1. *There is a two-party communication protocol using public randomness that computes* $\widetilde{\mathsf{GapHam}}$ *with error probability less than* $1/3$ *and total communication* $O(k \log k)$*.*

Proof. **Protocol.** Alice holds $(\mathbf{x}, \mathbf{u})$, and Bob holds $(\mathbf{y}, \mathbf{v})$.

I). Let $q = \lceil 18 \ln(6k) \rceil$. Using public randomness, the parties sample positions $I_{i,j}$ uniformly and independently from $[m]$, for $i \in [k]$ and $j \in [q]$. Positions may be sampled more than once.

II). Alice sends the kq bits $\mathbf{x}_i(I_{i,j})$ to Bob. Bob computes the sampled differences

$$\widetilde{\mathbf{z}}_i(j) = \mathbf{x}_i(I_{i,j}) \oplus \mathbf{y}_i(I_{i,j}) \qquad (i \in [k],\ j \in [q]).$$

III). Bob computes

$$h_i = \mathbf{1}\left\{\frac{1}{q}\sum_{j=1}^{q}\widetilde{\mathbf{z}}_i(j) \geq \frac{1}{2}\right\} \quad \forall i \in [k], \qquad h = 1 + \sum_{i=1}^{k} h_i 2^{k-i}.$$

He sends $h_1, \ldots, h_k$ to Alice, so both of them know the address $h \in [m]$.

IV). They follow the test in Theorem 3.7 to compute $\mathsf{Check}(\mathbf{z}, \mathbf{u}_h \oplus \mathbf{v}_h)$. Let p_A and p_B be the polynomials encoded by $\mathbf{u}_h$ and $\mathbf{v}_h$, respectively, and let $p = p_A + p_B$. Using public randomness independent of the sampled positions, they sample $r \in \mathbb{F}_{2^{4k}}$ uniformly. Alice sends

$$p_A(r), \qquad f_{(\mathbf{x},p_A)}(r), \qquad g_{(\mathbf{x},p_A)}(r), \qquad p_A(\tau_d).$$

Bob computes

$$\begin{aligned} p(r) &= p_A(r) + p_B(r), \\ f_{(\mathbf{z},p)}(r) &= f_{(\mathbf{x},p_A)}(r) + f_{(\mathbf{y},p_B)}(r) + f_{(\mathbf{0},p_0)}(r), \\ g_{(\mathbf{z},p)}(r) &= g_{(\mathbf{x},p_A)}(r) + g_{(\mathbf{y},p_B)}(r) + g_{(\mathbf{0},p_0)}(r), \\ p(\tau_d) &= p_A(\tau_d) + p_B(\tau_d), \end{aligned}$$

and outputs

$$b = \mathbf{1}\big\{p(\tau_d) = 0 \text{ and } p(r) = f_{(\mathbf{z},p)}(r) g_{(\mathbf{z},p)}(r)\big\}.$$

Correctness. Fix the entire input, and let $\mathbf{c}_j = \mathbf{u}_j \oplus \mathbf{v}_j$ for every $j \in [m]$. The only randomness is in the sampled positions and $r \sim \mathbb{F}_{2^{4k}}$. For every j with $\Pr[h = j] > 0$, conditioning on $h = j$ fixes the certificate $\mathbf{c}_j$ and leaves r uniform. Theorem 3.7 therefore gives

$$\Pr[b \neq \mathsf{Check}(\mathbf{z}, \mathbf{c}_j) \mid h = j] < \frac{1}{6}. \tag{4.1}$$

1). Suppose $\mathsf{Ind}(\mathbf{z}) = 1$. For each $i \in [k]$, the sampled bits $\widetilde{\mathbf{z}}_i(1), \ldots, \widetilde{\mathbf{z}}_i(q)$ are independent and have expectation $|\mathbf{z}_i|/m \in [0, 1/3] \cup [2/3, 1]$. The corresponding tail bound for independent bits gives

$$\Pr[h_i \neq \mathsf{GapHam}(\mathbf{x}_i, \mathbf{y}_i)] \leq \exp\left(-2q\left(\frac{1}{6}\right)^2\right) = e^{-q/18} \leq \frac{1}{6k}.$$

Hence,

$$\Pr[h \neq \mathsf{Ads}(\mathbf{z})] \leq \sum_{i=1}^{k} \Pr[h_i \neq \mathsf{GapHam}(\mathbf{x}_i, \mathbf{y}_i)] \leq \frac{1}{6}.$$

When $h = \mathsf{Ads}(\mathbf{z})$, Definition 3.8 gives $\widetilde{\mathsf{GapHam}} = \mathsf{Check}(\mathbf{z}, \mathbf{c}_h)$. Equation (4.1) bounds the probability of a wrong answer on this event by less than $1/6$. Consequently,

$$\Pr[b \neq \widetilde{\mathsf{GapHam}}] \leq \Pr[h \neq \mathsf{Ads}(\mathbf{z})] + \Pr[b \neq \widetilde{\mathsf{GapHam}},\ h = \mathsf{Ads}(\mathbf{z})] < \frac{1}{6} + \frac{1}{6} = \frac{1}{3}.$$

2). Suppose $\mathsf{Ind}(\mathbf{z}) = 0$. By Definition 3.8, $\widetilde{\mathsf{GapHam}} = 0$; and Lemma 3.6 gives

$$\mathsf{Check}(\mathbf{z}, \mathbf{c}_j) = 0 \qquad \text{for every } j \in [m].$$

Thus every possible address selects a certificate for which Check is 0. Applying (4.1) at each such address yields

$$\Pr[b \neq \widetilde{\mathsf{GapHam}}] = \sum_{j \in [m]} \Pr[h = j] \Pr[b = 1 \mid h = j] < \frac{1}{6} \sum_{j \in [m]} \Pr[h = j] = \frac{1}{6}.$$

Cost. Steps II and III use kq and k bits, respectively. In Step IV, Alice sends four field elements, each encoded by $4k$ bits. The total communication is $kq + k + 16k = O(k \log k)$. □

5 Rectangle Bounds

In this section, we show that any monochromatic rectangle of $\widetilde{\mathsf{GapHam}}$ has exponentially small density under the uniform distribution. Let the input space be

$$U = V = (\{0,1\}^{4k(2d-1)})^m, \qquad A = \{0,1\}^{km} \times U, \qquad B = \{0,1\}^{km} \times V.$$

Let μ_A and μ_B be the uniform measures on A and B, and $\mu = \mu_A \times \mu_B$. A rectangle is a direct product like $A' \times B'$ with $A' \subseteq A$ and $B' \subseteq B$. Its density is $\mu(A' \times B') := \mu_A(A')\mu_B(B')$. For any $b \in \{0,1\}$, we call $A' \times B'$ a b-rectangle if $\widetilde{\mathsf{GapHam}}$ equals b on every input in it. We will prove the following theorem.

Theorem 5.1. *Under the uniform measure μ, every monochromatic rectangle of $\widetilde{\mathsf{GapHam}}$ has density at most $2^{-m/(8k)}$.*

In view of our choice of the parameters, $m = 2^k, d = O(km^2)$, thus $m = \mathrm{poly}(n)$ for the input length n, establishing Theorem 1.1.

Before proving Theorem 5.1, we need the following two lemmas. We first show that two large sets of binary strings must contain a pair with small Hamming distance.

Lemma 5.2. *For any integer $d \geq 1$, let $E, Q \subseteq \{0,1\}^d$ be nonempty sets, and let*

$$\alpha := \frac{|E|}{2^d}, \qquad \beta := \frac{|Q|}{2^d}.$$

There exist $\mathbf{a} \in E$ and $\mathbf{b} \in Q$ such that

$$|\mathbf{a} \oplus \mathbf{b}| \leq \sqrt{\frac{d}{2} \ln \frac{1}{\alpha}} + \sqrt{\frac{d}{2} \ln \frac{1}{\beta}} \leq \sqrt{d \ln \frac{1}{\alpha\beta}}.$$

Proof. Let $\mathbf{Z}$ be uniform on $\{0,1\}^d$, and define

$$f(\mathbf{z}) = \min_{\mathbf{a} \in E} |\mathbf{z} \oplus \mathbf{a}|, \qquad s = \mathbb{E}[f(\mathbf{Z})].$$

The bits of $\mathbf{Z}$ are independent, and changing one bit changes f by at most 1. The bounded differences inequality therefore gives, for every $t \geq 0$,

$$\Pr[f(\mathbf{Z}) - s \leq -t] \leq e^{-2t^2/d}, \quad \Pr[f(\mathbf{Z}) - s \geq t] \leq e^{-2t^2/d}.$$

Since $f(\mathbf{z}) = 0$ exactly when $\mathbf{z} \in E$, the first inequality with $t = s$ gives

$$\alpha = \Pr[f(\mathbf{Z}) = 0] \leq e^{-2s^2/d}.$$

Taking logarithms yields

$$s \leq \sqrt{\frac{d}{2} \ln \frac{1}{\alpha}}.$$

Choose $\mathbf{b} \in Q$ minimizing $f(\mathbf{b})$, and choose $\mathbf{a} \in E$ such that $|\mathbf{a} \oplus \mathbf{b}| = f(\mathbf{b})$. If $f(\mathbf{b}) \leq s$, the first claimed bound follows immediately. If $f(\mathbf{b}) > s$, every $\mathbf{z} \in Q$ satisfies $f(\mathbf{z}) \geq f(\mathbf{b})$, so the second tail inequality gives

$$\beta \leq \Pr[f(\mathbf{Z}) \geq f(\mathbf{b})] \leq e^{-2(f(\mathbf{b})-s)^2/d}.$$

Taking logarithms again yields

$$f(\mathbf{b}) - s \le \sqrt{\frac{d}{2} \ln \frac{1}{\beta}}.$$

Adding the bounds on s and $f(\mathbf{b}) - s$ proves the first claimed inequality. The inequality $(\sqrt{a}+\sqrt{b})^2 \le 2(a+b)$ for $a, b \ge 0$ proves the second one. □

We next show that a large set of tables contains more than half of all possible vectors at most addresses.

Lemma 5.3. *Let $\mathcal{U} \subseteq U$ be nonempty, and set $\sigma = |\mathcal{U}|/2^{m \cdot 4k(2d-1)}$. For each address $h \in [m]$, define the set of vectors that occur at that address by $\pi_h(\mathcal{U}) = \{\mathbf{u}_h : \mathbf{u} \in \mathcal{U}\}$. Then at most $\log(1/\sigma)$ addresses contain at most half of all possible vectors:*

$$\left| \{h \in [m] : |\pi_h(\mathcal{U})| \le 2^{4k(2d-1)-1}\} \right| \le \log(1/\sigma).$$

Proof. Let

$$S = \{h \in [m] : |\pi_h(\mathcal{U})| \le 2^{4k(2d-1)-1}\}.$$

Every table in $\mathcal{U}$ chooses its vector at address h from $\pi_h(\mathcal{U})$. Thus

$$|\mathcal{U}| \le \prod_{h=1}^{m} |\pi_h(\mathcal{U})| \le 2^{m \cdot 4k(2d-1) - |S|}.$$

Dividing by $2^{m \cdot 4k(2d-1)}$ gives $\sigma \le 2^{-|S|}$. □

5.1 Proof of Theorem 5.1

We use Lemma 5.2 and Lemma 5.3 to rule out large 0-rectangles, and the uniqueness statement in Lemma 3.6 to bound the density of 1-rectangles.

0-rectangles.

Suppose a 0-rectangle $A' \times B'$ has

$$\alpha = \mu_A(A'), \qquad \beta = \mu_B(B'), \qquad \alpha\beta \ge 2^{-m/(8k)}.$$

We will find a 1-input in $A' \times B'$, yielding a contradiction. This argument has four steps.

Step 1. We first keep those $\mathbf{x}$ and $\mathbf{y}$ that allow many choices of their tables. For each $\mathbf{x}, \mathbf{y} \in \{0,1\}^{km}$, define the corresponding tables as

$$\mathcal{U}_{\mathbf{x}} = \{\mathbf{u} \in U : (\mathbf{x}, \mathbf{u}) \in A'\}, \qquad \mathcal{V}_{\mathbf{y}} = \{\mathbf{v} \in V : (\mathbf{y}, \mathbf{v}) \in B'\}.$$

Define their density by

$$\alpha_{\mathbf{x}} = \frac{|\mathcal{U}_{\mathbf{x}}|}{2^{m \cdot 4k(2d-1)}}, \qquad \beta_{\mathbf{y}} = \frac{|\mathcal{V}_{\mathbf{y}}|}{2^{m \cdot 4k(2d-1)}}.$$

Define the strings for which these densities are at least half:

$$X = \{\mathbf{x} : \alpha_{\mathbf{x}} \ge \alpha/2\}, \qquad Y = \{\mathbf{y} : \beta_{\mathbf{y}} \ge \beta/2\}.$$

Considering the two cases X and X^c, we have

$$\alpha = \sum_{\mathbf{x}} 2^{-km} \cdot \alpha_{\mathbf{x}} \le \frac{|X|}{2^{km}} + \frac{\alpha}{2}.$$

The same argument applies to Y, so

$$\frac{|X|}{2^{km}} \ge \frac{\alpha}{2}, \qquad \frac{|Y|}{2^{km}} \ge \frac{\beta}{2}.$$

Step 2. We find one address where many inputs on each side allow more than half of all possible entries. For each $\mathbf{x} \in X$ and $\mathbf{y} \in Y$, let

$$S_{\mathbf{x}} = \{h \in [m] : |\pi_h(\mathcal{U}_{\mathbf{x}})| \le 2^{4k(2d-1)}/2\}, \quad T_{\mathbf{y}} = \{h \in [m] : |\pi_h(\mathcal{V}_{\mathbf{y}})| \le 2^{4k(2d-1)}/2\}.$$

Lemma 5.3 gives

$$|S_{\mathbf{x}}| \le \log \frac{1}{\alpha_{\mathbf{x}}} \le \log \frac{2}{\alpha}, \qquad |T_{\mathbf{y}}| \le \log \frac{1}{\beta_{\mathbf{y}}} \le \log \frac{2}{\beta}.$$

Define the addresses that occur in these sets for more than half of X or Y by

$$S = \{h \in [m] : |\{\mathbf{x} \in X : h \in S_{\mathbf{x}}\}| > |X|/2\}, \quad T = \{h \in [m] : |\{\mathbf{y} \in Y : h \in T_{\mathbf{y}}\}| > |Y|/2\}.$$

Counting the pairs $(h, \mathbf{x})$ with $h \in S_{\mathbf{x}}$ in two different ways gives

$$\frac{|S|\,|X|}{2} \le \sum_{h=1}^{m} |\{\mathbf{x} \in X : h \in S_{\mathbf{x}}\}| = \sum_{\mathbf{x} \in X} |S_{\mathbf{x}}| \le |X| \log \frac{2}{\alpha}.$$

Thus $|S| \le 2\log(2/\alpha)$; counting the pairs $(h, \mathbf{y})$ in the same way bounds $|T| \le 2\log(2/\beta)$. Consequently,

$$|S \cup T| \le 2\log \frac{2}{\alpha} + 2\log \frac{2}{\beta} = 2\log \frac{1}{\alpha\beta} + 4 \le \frac{m}{4k} + 4 < m.$$

So there exists an address $h \in [m] \setminus (S \cup T)$. Define

$$X_h = \{\mathbf{x} \in X : h \notin S_{\mathbf{x}}\}, \qquad Y_h = \{\mathbf{y} \in Y : h \notin T_{\mathbf{y}}\}.$$

Since $h \notin S \cup T$,

$$\big|\{x \in X : h \in S_x\}\big| \le |X|/2, \qquad \big|\{y \in Y : h \in T_y\}\big| \le |Y|/2.$$

Hence

$$\frac{|X_h|}{2^{km}} \ge \frac{|X|}{2^{km+1}} \ge \frac{\alpha}{4}, \qquad \frac{|Y_h|}{2^{km}} \ge \frac{|Y|}{2^{km+1}} \ge \frac{\beta}{4}.$$

Moreover, their definitions give, for every $\mathbf{x} \in X_h$ and $\mathbf{y} \in Y_h$,

$$|\pi_h(\mathcal{U}_{\mathbf{x}})| > 2^{4k(2d-1)-1}, \qquad |\pi_h(\mathcal{V}_{\mathbf{y}})| > 2^{4k(2d-1)-1}.$$

Step 3. We choose $\mathbf{x}, \mathbf{y}$ with $\mathsf{Ind}(\mathbf{x} \oplus \mathbf{y}) = 1$ and $\mathsf{Ads}(\mathbf{x} \oplus \mathbf{y}) = h$. Write $h = 1 + \sum_{i=1}^{k} h_i 2^{k-i}$, where $h_i \in \{0, 1\}$. Let $\mathbf{1}_m$ be the all-1 vector of length m. Adding the fixed vector $(h_1\mathbf{1}_m, \ldots, h_k\mathbf{1}_m)$ by XOR preserves the size of Y_h. Lemma 5.2, applied to X_h and the set

$$\{\mathbf{y} \oplus (h_1\mathbf{1}_m, \ldots, h_k\mathbf{1}_m) : \mathbf{y} \in Y_h\},$$

therefore gives $\mathbf{x} \in X_h$ and $\mathbf{y} \in Y_h$ such that the distance between $\mathbf{x}$ and $\mathbf{y} \oplus (h_1\mathbf{1}_m, \ldots, h_k\mathbf{1}_m)$ is

$$|\mathbf{x} \oplus \mathbf{y} \oplus (h_1\mathbf{1}_m, \ldots, h_k\mathbf{1}_m)| \leq \sqrt{km \ln \frac{16}{\alpha\beta}} \leq m\sqrt{\left(\frac{1}{8} + \frac{4k}{m}\right) \ln 2} \leq \frac{m}{3}. \tag{5.1}$$

So each block contributes at most $m/3$. If $h_i = 0$, $|\mathbf{x}_i \oplus \mathbf{y}_i| < m/3$; if $h_i = 1$, $|\mathbf{x}_i \oplus \mathbf{y}_i| > 2m/3$. The definitions of Ind and Ads give

$$\mathsf{Ind}(\mathbf{x} \oplus \mathbf{y}) = 1, \qquad \mathsf{Ads}(\mathbf{x} \oplus \mathbf{y}) = h.$$

Step 4. We choose tables that give output 1 inside $A' \times B'$, contradicting the assumed output 0. Lemma 3.6 gives a certificate $\mathbf{c}^*$ such that

$$\mathsf{Check}(\mathbf{x} \oplus \mathbf{y}, \mathbf{c}^*) = 1.$$

By Step 2, more than half of the vectors in $\{0,1\}^{4k(2d-1)}$ occur in $\pi_h(\mathcal{U}_\mathbf{x})$. The set

$$\{\mathbf{v}_h \oplus \mathbf{c}^* : \mathbf{v}_h \in \pi_h(\mathcal{V}_\mathbf{y})\}$$

also contains more than half of these vectors, since XOR with $\mathbf{c}^*$ doesn't change the size. If these two sets were disjoint, their union would contain more than $2^{4k(2d-1)}$ vectors in a space of that size. Therefore, they intersect.

By the definition of π_h, there exist tables $\mathbf{u} \in \mathcal{U}_\mathbf{x}$ and $\mathbf{v} \in \mathcal{V}_\mathbf{y}$ such that

$$\mathbf{u}_h \oplus \mathbf{v}_h = \mathbf{c}^*.$$

The definitions of $\mathcal{U}_\mathbf{x}$ and $\mathcal{V}_\mathbf{y}$ ensure that

$$((\mathbf{x}, \mathbf{u}), (\mathbf{y}, \mathbf{v})) \in A' \times B'.$$

Since $\mathsf{Ind}(\mathbf{x} \oplus \mathbf{y}) = 1$ and $\mathsf{Ads}(\mathbf{x} \oplus \mathbf{y}) = h$, Definition 3.8 gives

$$\widetilde{\mathsf{GapHam}}(\mathbf{x}, \mathbf{u}; \mathbf{y}, \mathbf{v}) = \mathsf{Check}(\mathbf{x} \oplus \mathbf{y}, \mathbf{c}^*) = 1.$$

This contradicts the assumption that $A' \times B'$ is a 0-rectangle.

1-rectangles.

We bound the density of all 1-inputs. Fix $\mathbf{x}$ and $\mathbf{y}$. If $\mathsf{Ind}(\mathbf{x} \oplus \mathbf{y}) = 0$, then $\widetilde{\mathsf{GapHam}} = 0$ for all tables. If $\mathsf{Ind}(\mathbf{x} \oplus \mathbf{y}) = 1$, the address $h = \mathsf{Ads}(\mathbf{x} \oplus \mathbf{y})$ is fixed. Lemma 3.6 gives exactly one certificate $\mathbf{c}^*$ for which $\mathsf{Check}(\mathbf{x} \oplus \mathbf{y}, \mathbf{c}^*) = 1$. By Definition 3.8, the output equals 1 precisely when

$$\mathbf{u}_h \oplus \mathbf{v}_h = \mathbf{c}^*.$$

Under the uniform distribution on $U \times V$, the vectors $\mathbf{u}_h$ and $\mathbf{v}_h$ are independent and uniform on $\{0,1\}^{4k(2d-1)}$. For every choice of $\mathbf{u}_h$, exactly one of the $2^{4k(2d-1)}$ choices of $\mathbf{v}_h$ satisfies this equation. Thus, in both cases for $\mathsf{Ind}(\mathbf{x} \oplus \mathbf{y})$,

$$\Pr_{\mathbf{u},\mathbf{v}}[\widetilde{\mathsf{GapHam}}(\mathbf{x}, \mathbf{u}; \mathbf{y}, \mathbf{v}) = 1] \leq 2^{-4k(2d-1)}.$$

For any 1-rectangle $A' \times B'$, every input in it has output 1. Averaging over $\mathbf{x}$ and $\mathbf{y}$ therefore gives

$$\mu(A' \times B') \leq \Pr_{\mu}[\widetilde{\mathsf{GapHam}} = 1] \leq 2^{-4k(2d-1)} < 2^{-m/(8k)}.$$

Acknowledgement

The authors used GPT-6 for exploratory analysis, idea testing, editorial polishing. The authors reviewed and finalized all mathematical claims and proofs, and take full responsibility for the correctness, exposition, and attribution in the final manuscript.

References

[ABBD+16] Anurag Anshu, Aleksandrs Belovs, Shalev Ben-David, Mika Göös, Rahul Jain, Robin Kothari, Troy Lee, and Miklos Santha. Separations in communication complexity using cheat sheets and information complexity. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 555–564. IEEE, 2016.

[ABDK16] Scott Aaronson, Shalev Ben-David, and Robin Kothari. Separations in query complexity using cheat sheets. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 863–876, 2016.

[BBCG+19] Dan Boneh, Elette Boyle, Henry Corrigan-Gibbs, Niv Gilboa, and Yuval Ishai. Zero-knowledge proofs on secret-shared data via fully linear pcps. In *Annual international cryptology conference*, pages 67–97. Springer, 2019.

[BFS86] László Babai, Péter Frankl, and János Simon. Complexity classes in communication complexity theory. In *27th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 337–347. IEEE, 1986.

[CLV19] Arkadev Chattopadhyay, Shachar Lovett, and Marc Vinyals. Equality alone does not simulate randomness. In *34th Computational Complexity Conference (CCC 2019)*, volume 137 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 14:1–14:11. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2019.

[Gav26] Dmytro Gavinsky. On the quantum communication complexity of total functions. arXiv:2608.18784v1, `https://arxiv.org/abs/2608.18784v1`, 2026. Preliminary manuscript, August 19, 2026.

[GKPW17] Mika Göös, Pritish Kamath, Toniann Pitassi, and Thomas Watson. Query-to-communication lifting for P^{NP}. In *32nd Computational Complexity Conference (CCC 2017)*, volume 79 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 12:1–12:16. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2017.

[GPW18] Mika Göös, Toniann Pitassi, and Thomas Watson. The landscape of communication complexity classes. *Computational Complexity*, 27(2):245–304, 2018. Preliminary version in ICALP 2016.

[HH24] Hamed Hatami and Pooya Hatami. Guest column: Structure in communication complexity and constant-cost complexity classes. *ACM SIGACT News*, 55(1):67–93, 2024.

[IW10] Russell Impagliazzo and Ryan Williams. Communication complexity with synchronized clocks. In *25th Annual IEEE Conference on Computational Complexity (CCC)*, pages 259–269. IEEE, 2010.

[NW95] Noam Nisan and Avi Wigderson. On rank vs. communication complexity. *Combinatorica*, 15(4):557–565, 1995.

[PSS14] Periklis Papakonstantinou, Dominik Scheder, and Hao Song. Overlays and limited memory communication. In *29th Annual IEEE Conference on Computational Complexity (CCC)*, pages 298–308. IEEE, 2014.

[PSW20] Toniann Pitassi, Morgan Shirley, and Thomas Watson. Nondeterministic and randomized boolean hierarchies in communication complexity. In *47th International Colloquium on Automata, Languages, and Programming (ICALP 2020)*, volume 168 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 92:1–92:19. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020.

[Yao79] Andrew Chi-Chih Yao. Some complexity questions related to distributive computing (preliminary report). In *11th Annual ACM Symposium on Theory of Computing (STOC)*, pages 209–213. ACM, 1979.

A Auxiliary Proofs

Proof of Lemma 3.6. Run the circuit on $\mathbf{z}$, and let a_v^*, b_v^* be the actual inputs to its v-th AND gate.

Part 1), existence. Suppose $\mathsf{Ind}(\mathbf{z}) = 1$. Define

$$p^*(T) = \left(\sum_{v=1}^{d} \delta_v(T) a_v^*\right)\left(\sum_{v=1}^{d} \delta_v(T) b_v^*\right).$$

This polynomial has degree at most $2d-2$, so its coefficients define a certificate. At each τ_v,

$$p^*(\tau_v) = a_v^* b_v^*.$$

We prove by induction on v that using these claimed outputs preserves the actual inputs and output of each AND gate. Before the first such gate, the circuit uses only its original inputs, constants, NOT, and XOR, so both calculations agree. If the earlier AND outputs agree, applying the same NOT and XOR operations gives

$$a_v(\mathbf{z}, p^*) = a_v^*, \qquad b_v(\mathbf{z}, p^*) = b_v^*.$$

The claimed output $p^*(\tau_v) = a_v^* b_v^*$ is also the actual output, completing the induction. The definitions of f and g now give

$$p^* \equiv f_{(\mathbf{z},p^*)} g_{(\mathbf{z},p^*)}.$$

The final gate outputs $1 - \mathsf{Ind}(\mathbf{z})$, so

$$p^*(\tau_d) = 1 - \mathsf{Ind}(\mathbf{z}) = 0.$$

Both requirements hold, and hence $\mathsf{Check}(\mathbf{z}, p^*) = 1$.

Part 1), uniqueness. Suppose $\mathsf{Check}(\mathbf{z}, p) = 1$. For every $v \in [d]$, the polynomial identity implies

$$p(\tau_v) = f_{(\mathbf{z},p)}(\tau_v) g_{(\mathbf{z},p)}(\tau_v) = a_v(\mathbf{z}, p) b_v(\mathbf{z}, p).$$

Again, the inputs to the first AND agree with their actual values. If the earlier AND outputs agree with their actual values, the intervening NOT and XOR operations give $a_v(\mathbf{z}, p) = a_v^*$ and

$b_v(\mathbf{z}, p) = b_v^*$. The displayed equality then gives $p(\tau_v) = a_v^* b_v^*$, so the output also agrees. Induction on v therefore proves these equalities for all $v \in [d]$. Consequently,

$$p \equiv f_{(\mathbf{z},p)} g_{(\mathbf{z},p)} \equiv \left(\sum_{v=1}^{d} \delta_v a_v^* \right) \left(\sum_{v=1}^{d} \delta_v b_v^* \right) \equiv p^*.$$

Thus all coefficients of p are fixed, and the certificate is unique.

Part 2). Suppose $\mathsf{Ind}(\mathbf{z}) = 0$ and $\mathsf{Check}(\mathbf{z}, p) = 1$. The induction in the uniqueness argument uses only $p \equiv f_{(\mathbf{z},p)} g_{(\mathbf{z},p)}$, so it still shows that every claimed AND output equals its actual output. In particular,

$$p(\tau_d) = 1 - \mathsf{Ind}(\mathbf{z}) = 1.$$

This contradicts the requirement $p(\tau_d) = 0$. Therefore no certificate has $\mathsf{Check}(\mathbf{z}, \mathbf{c}) = 1$. □

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il